COMPUTATION OF FORCES IN STRONGLY NONLINEAR MAGNETIC FIELDS USING HIGHER-ORDER EGGSHELL ALGORITHM

Karban, P.[1], Korous, L.[1], Mach, F .[1], Doležel, I.[2]
[1] University of West Bohemia, Pilsen, Czech Republic, {karban, korous, fmach}@kte.zcu.cz
[2] Czech Technical University, Prague, Czech Republic, dolezel@fel.cvut.cz

***Annotation* — A novel version of the eggshell-based procedure for numerical computation of magnetic forces and torques acting on ferromagnetic bodies in highly nonlinear magnetic fields is presented. The procedure works with a fully adaptive higher-order finite element method developed for years in our research group, that is implemented in own code Agros2D and library Hermes. The power of the methodology and both codes is demonstrated on the solution of two typical examples: computation of the static characteristic of a magnetic actuator and torque characteristic of a flux-switched permanent-magnet machine. The results obtained are compared with data calculated by several other available codes.**

***Keywords*: nonlinear magnetic field, force and torque effects, higher-order finite element method, *hp*-adaptivity, eggshell method, numerical analysis.**

INTRODUCTION

An integral part of modeling numerous electrical devices containing movable parts is determination of their static or dynamic characteristics representing the dependences of the forces or torques acting on such parts on their current position.

The examples abound. Mentioned can be, for example, electromagnetic actuators, rotating machines, some types of levitation devices and many others.

Evaluation of the force and torque effects in low magnetic fields is not so much complicated and existing techniques (Virtual Work, Maxwell Stress Tensor or Eggshell Method) implemented in available professional codes provide the results with a sufficient accuracy. In strong magnetic fields, however, their determination is still a challenge. The computations are often very slow, convergence is rather poor and accuracy of the results is insufficient (the convergence processes may be accompanied by oscillations and other undesirable phenomena) [1–3].

The authors offer an alternative approach based on the eggshell algorithm that works with the fully adaptive higher-order finite element method. This method working with polynomials of higher orders brings a high level of smoothness to the solution (the degrees of polynomials may change arbitrarily according to the adaptive algorithms), which suppresses peaks and oscillations in the resultant curves. The complete methodology is implemented into own code Agros2D developed for years in our research group and intended for the solution of 2D nonlinear and nonstationary multiphysics problems.

The goal of the paper is to demonstrate the power of the approach on two typical examples: computation of the static characteristics of a magnetic actuator and torque characteristic of a flux-switched permanent-magnet (FSPM) machine.

MATHEMATICAL MODEL OF THE PROBLEM

The mathematical model of the problem consists of the partial differential equation describing the distribution of magnetic field area and an appropriate expression for determining the force or torque.

In case of 2D low-frequency problems, the distribution of magnetic field may conveniently be described by the parabolic equation for the magnetic vector potential $\boldsymbol{A}$ in the form [4]

$$\mathrm{curl}\left(\frac{1}{\mathrm{M}}\mathrm{curl}\,\boldsymbol{A}\right)+\Gamma\left(\frac{\partial \boldsymbol{A}}{\partial t}-\boldsymbol{v}\times\mathrm{curl}\,\boldsymbol{A}\right)=\boldsymbol{J}_{ext}\,, \qquad (1)$$

where M is the magnetic permeability, Γ denotes the electric conductivity, $\boldsymbol{v}$ is the vector of velocity of the movable part, and $\boldsymbol{J}_{ext}$ stands for the vector of the external current density in the field currents. As far as the field currents do not vary in time and velocity $\boldsymbol{v}$ is sufficiently low, the second term on the left-hand side can be neglected and (1) reduces to the form

$$\mathrm{curl}\left(\frac{1}{\mathrm{M}}\mathrm{curl}\,\boldsymbol{A}\right)=\boldsymbol{J}_{ext}\,, \qquad (2)$$

that is often used, for example, for solving magnetic fields in DC actuators and other similar devices.

The force acting on the movable part may be determined in two different ways. The first one is the Method of Virtual Work. Here, the local force $\boldsymbol{f}_m$ acting at a given point is given by the formula [5]

$$\boldsymbol{f}_m=-\mathrm{grad}\ w_m\,, \qquad (3)$$

where w_m is the local density of magnetic field energy. The total force $\boldsymbol{F}_m$ acting on the body is obtained by integration of (3) over the volume V of the body. This provides

$$\boldsymbol{F}_m=-\mathrm{grad}\ W_m\,, \qquad (4)$$

where W_m is the total magnetic energy connected with the body. Analogously, the total torque $\boldsymbol{T}_m$ acting on the body follows from the expression

$$\boldsymbol{T}_m = -\int_V \boldsymbol{r} \times \operatorname{grad} w_m \, \mathrm{d}V , \qquad (5)$$

where $\boldsymbol{r}$ is the position vector.

The local and total energies of magnetic field are (in case of nonlinear environment) given by the formulas

$$w_m = \int_0^{\boldsymbol{B}} \boldsymbol{H} \mathrm{d}\boldsymbol{B}, \quad W_m = \int_V w_m \, \mathrm{d}V \qquad (6)$$

where $\boldsymbol{H}$ and $\boldsymbol{B}$ are the field vectors.

But computation of quantities $\boldsymbol{F}_m$ (4) or $\boldsymbol{T}_m$ (5) is not too much advantageous. The determination of $\operatorname{grad} W_m$ or $\operatorname{grad} w_m$ requires calculating the relevant energies in two near positions (thus, two computations of the distribution of magnetic field). For two close positions the corresponding values of energy differ only slightly. The numerical computation of the gradients is then often burdened by unacceptable errors. This method of computation is, therefore, slowly abandoned and professional codes do not contain it.

Another method is based on the Maxwell Stress Tensor. This tensor $\boldsymbol{\sigma}_m$ may formally be written in the form

$$\boldsymbol{\sigma}_m = -\frac{1}{2\mu}\left(\boldsymbol{B} \cdot \boldsymbol{B}\right)\boldsymbol{I} + \frac{1}{\mu}\boldsymbol{B} \otimes \boldsymbol{B} . \qquad (7)$$

Here, μ stands for the permeability, sign $\otimes$ denotes the dyadic product whose result is a matrix and $\boldsymbol{I}$ is the unit diagonal matrix.

The total force $\boldsymbol{F}_m$ acting on the body is now given by the integral

$$\boldsymbol{F}_m = \oint_S \boldsymbol{\sigma}_m \, \mathrm{d}\boldsymbol{S} \qquad (8)$$

where $\boldsymbol{S}$ denotes the outward unit normal vector to the boundary S of the movable part. In a similar way we can also write the formula for the mechanical torque

$$\boldsymbol{T}_m = \oint_S \boldsymbol{r} \times \left(\boldsymbol{\sigma}_m \, \mathrm{d}\boldsymbol{S}\right) . \qquad (9)$$

Knowledge of the Maxwell Stress Tensor is also the starting point for the Eggshell Method. The Eggshell method [6] is based on introducing a thin shell over the surface of the body (see Pic. 1) and an appropriate function γ satisfying the conditions $\gamma = 0$ along the external boundary of the shell and $\gamma = 1$ along its internal boundary.

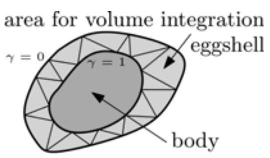

Fig. 1. Principal scheme of Eggshell method

Now the magnetic force acting on the body is

$$\boldsymbol{F}_m = \int_V \boldsymbol{\sigma}_m \cdot \operatorname{grad} \gamma \, \mathrm{d}V , \qquad (10)$$

where V is the volume of the eggshell. Function γ can be chosen in more ways. In our implementation, we use the solution of the Laplace equation in the eggshell domain obtained using higher-order finite elements. Since the eggshell domain is usually covered with very few elements (see Pic. 2), the additional computational costs caused by increasing the element order are quite negligible.

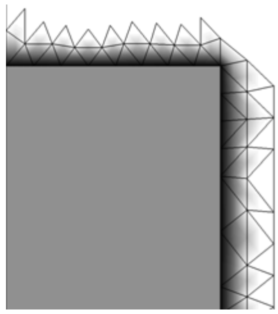
Fig. 2. Part of eggshell around a ferromagnetic body

NUMERICAL SOLUTION

The magnetic forces in 2D arrangements are solved numerically by our own code Agros2D [7], which is a powerful user's interface serving for pre-processing and post-processing of the problems solved. The code collaborates with the library Hermes [8] containing the most advanced algorithms for a fully adaptive solution of systems of generally nonlinear and non-stationary second-order partial differential equations (PDEs) based on the finite element method of higher order of accuracy.

Both codes written in C++ are freely distributable under the GNU General Public License. The codes are characterized by quite unique features such as [9]

- Solution of the system of PDEs is carried out monolithically, which means that the resultant numerical scheme is characterized by just one stiffness matrix. The PDEs are first rewritten into the weak forms whose numerical integration provides its coefficients. The integration is performed using the Gauss quadrature formulas.
- Fully automatic *hp*-adaptivity. When adaptivity is required, in every iteration step the solution is compared with the reference solution (realized on an approximately twice finer mesh), and the distribution of error is then used for selection of candidates for adaptivity. Based on sophisticated and subtle algorithms the adaptivity is realized either by a subdivision of the candidate element (*h*-adaptivity), by its description by a polynomial of a higher order (*p*-adaptivity), or by a combination of both above options (*hp*-adaptivity).
- Each physical field can be solved on quite a different mesh that best corresponds to its particulars. This is of great importance, for instance, for respecting skin effect in the magnetic field, while the temperature field is usually smooth. Special powerful higher-order techniques of mapping are then used to avoid any

numerical errors in the process of assembly of the stiffness matrix.

- In non-stationary processes every mesh can change in time, in accordance with the real evolution of the corresponding physical quantities.
- Easy treatment of the hanging nodes appearing on the boundaries of subdomains whose elements have to be refined. Usually, these nodes bring about a considerable increase of the number of the degrees of freedom (DOFs). The code contains higher-order algorithms for respecting these nodes without any need of an additional refinement of the external parts neighboring with the refined subdomain.
- Curved elements able to replace curvilinear parts of any boundary by a system of circular or elliptic arcs. These elements mostly allow reaching highly accurate results near the curvilinear boundaries with very low numbers of the DOFs.

ILLUSTRATIVE EXAMPLES

The power of the code and also of the method itself is demonstrated on two examples. The first one is computation of the static characteristic of a cylindrical magnetic actuator, the second example is the torque characteristic of a flux-switched permanent-magnet machine.

EXAMPLE 1

The principal arrangement of the cylindrical DC actuator together with the principal dimensions (in mm) is in Pic. 3. Its magnetic parts are made of Czech carbon steel ČSN 12 040 whose saturation characteristic is shown in Pic. 4.

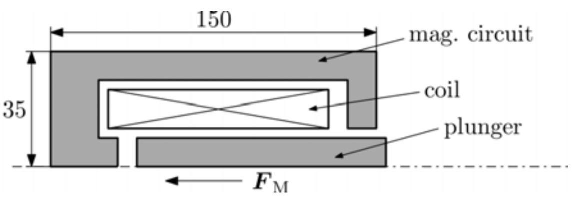

Fig. 3. Principal scheme of DC electromagnet

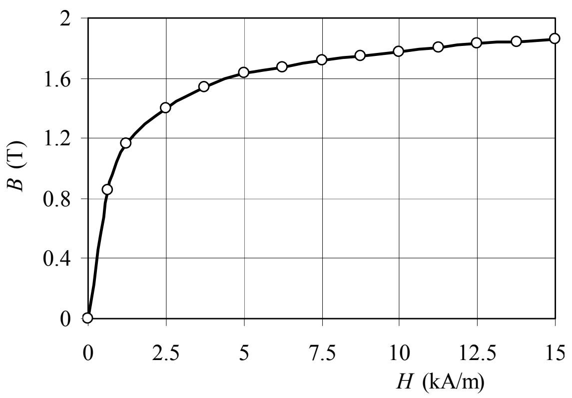

Fig. 4. Saturation curve of carbon steel ČSN 12 040

The coil containing 1200 turns is supplied by direct current $I = 3\,\mathrm{A}$, which is enough for reaching a high saturation of the magnetic circuit.

Table 1 contains the parameters of computation needed for determining the forces at three positions of the plunger (0.01, 0.02 and 0.03 m from the left end), i.e., in highly saturated states of the device. The number of DOFs was comparable, the prescribed residual norm was 10^{-4}.

Pic. 5 compares several methods of calculation of the static characteristic with measurement on the experimental device. The computations are performed for lower (but again mutually comparable) numbers of DOFs than in Tab. 1. Finally, Pic. 6 shows the static characteristic computed by Agros2D using the Maxwell tensor and two (first-order and third-order) Eggshell methods. It is clear that even the results obtained by the first-order Eggshell method are much better (the curve is smoother and better corresponds to the measured data).

Table 1

Comparison of parameters of computation using COMSOL and Agros2D for three positions of the plunger

code	COMSOL			Agros2D		
position (m)	0.01	0.02	0.03	0.01	0.02	0.03
DOFs	27938	28121	28046	27208	27529	28876
time (s)	73	153	35	10	19	9
iterations	132	275	62	21	27	20
force (N)	14.41	10.36	8.76	13.23	10.09	9.66

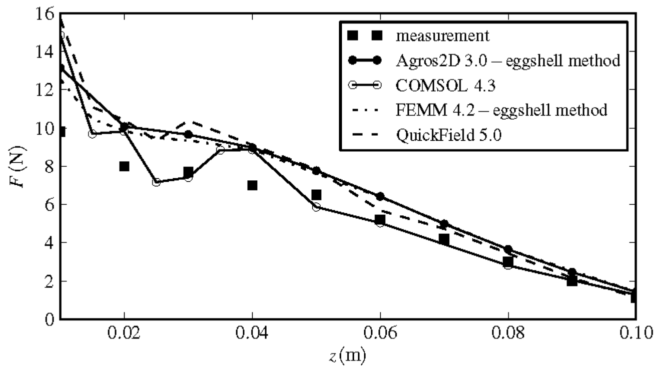

Fig. 5. Comparison of several methods for calculating force acting on armature of electromagnetic actuator

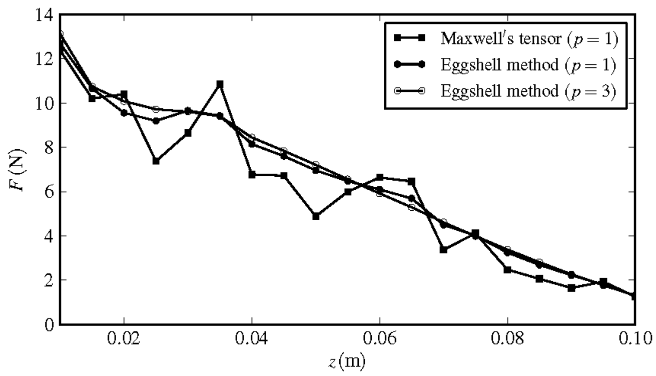

Fig. 6. Comparison of first-order Maxwell stress tensor, first-order eggshell method and third-order eggshell method

EXAMPLE 2

The arrangement of the flux-switched permanent-magnet machine together with its principal dimensions in mm is depicted in Pic. 7. It consists of two permanent magnets, two field coils, stator with four poles and a five-pole rotor.

The ferromagnetic parts are manufactured again from the Czech carbon steel ČSN 12 040. The permanent magnets VMM10 are of NdFeB type and their manufacturer provides the following parameters: remanence B_r = 1.45 T, relative permeability in the second quadrant μ_r = 1.21 and maximum allowable temperature is 80 °C.

The coils (whose purpose is to control the motion of the rotor) work sequentially and during the observed period the upper one carries direct current of density $J = 2.2 \times 10^6$ A/m^2. The lower coil is without current.

Pic. 8 shows the distribution of the magnetic torque acting on the rotor within the range of angles 0–80° (the angular distance between the axes of two neighboring poles on the rotor being 72°). While the results obtained from the virtual work and Maxwell stress tensor exhibit very distinct peaks (even more than 40 %), the third-order eggshell method provides a smooth curve. The measured points were obtained statically (rotor being in rest).

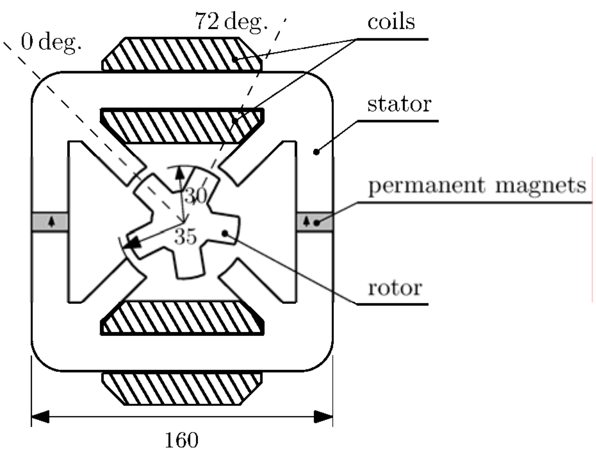

Fig. 7. Principal scheme of the FSPM machine

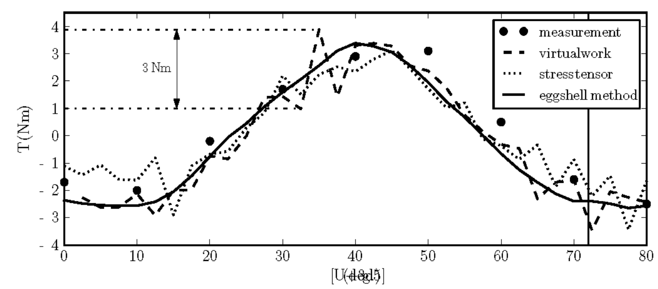

Fig. 8. Magnetic torque T_m acting on rotor versus angle ц calculated by various methods and its comparison with measurement

It should be noted that the convergence of the force or torque at one position of the rotor is mostly quite uniform (i.e., in the course of the iterative process its value grows or decreases monotonically to the correct value). But when using the method of Virtual Work or Maxwell Stress Tensor, their values in different positions are burdened with errors, which is the principal cause of the saw-like curve.

CONCLUSION

The computation of magnetic forces and torques in the presented (and also several other) arrangements by the above algorithm confirms its excellent properties. The only disadvantage is that this approach is still confined to 2D problems.

ACKNOWLEDGMENT

This work was supported by the European Regional Development Fund and Ministry of Education, Youth and Sports of the Czech Republic (project No. CZ.1.05/2.1.00/03.0094: Regional Innovation Center for Electrical Engineering - RICE) and by the project P102/11/0498 (Grant Agency of the Czech Republic).

BIBLIOGRAPHY

1. Zhu, Z. Q. An Accurate Subdomain Model for Magnetic Field Computation in Slotted Surface-Mounted Permanent-Magnet Machines. IEEE Trans. Magn. 46 (2010), No. 4, pp. 1100–1115.
2. Bachovkin, K. D., Hoburg, J. F., Post. R. F. Magnetic Fields and Forces in Permanent Magnet Levitated Bearings. IEEE Trans. Magn. 48 (2012), No. 7, pp. 2112–2120.
3. Valavi, M., Nysveen, A., Nilssen, R., Lorenz, B., Rolvag, T. Influence of Pole and Slot Combinations on Magnetic Forces and Vibration in Low-Speed PM Wind Generators. IEEE Trans. Magn. 49 (2012), No. 7, pp. 2112–2120. Accepted to IEEE Trans. Magn. 50 (2014).
4. Kuczmann, M., Ivanyi, A. The Finite Element Method in Magnetics. Akademiai Kiado, Budapest, 2008.
5. Stratton, J. A. Electromagnetic Theory, Wiley-IEEE Press, 2007.
6. Henrotte, F.,Felden, M, vander Giet, M., Hameyer, K. The Eggshell Approach for the Computation of Electromagnetic Forces in 2D and 3D. COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 23 (2004), No. 4, pp. 996–1005.
7. URL: agros2d.org.
8. URL: hpfem.org.
9. Karban, P., Mach, F.,Kůs, P., Pánek, D., Doležel, I. Numerical Solution of Coupled Problems Using Code Agros2D. Computing 95 (2013), No. 1, Supplement, pp 381–408, DOI 10.1007/s00607-013-0294-4.